

# Taking back control of privacy: a novel framework for preserving cloud-based firewall policy confidentiality

Tytus Kurek<sup>1</sup> · Marcin Niemiec<sup>1</sup> · Artur Lason<sup>1</sup>

Published online: 26 May 2015

© The Author(s) 2015. This article is published with open access at Springerlink.com

**Abstract** As the cloud computing paradigm evolves, new types of cloud-based services have become available, including security services. Some of the most important and most commonly adopted security services are firewall services. These cannot be easily deployed in a cloud, however, because of a lack of mechanisms preserving firewall policy confidentiality. Even if they were provided, the customer traffic flowing through the Cloud Service Provider infrastructure would still be exposed to eavesdropping and information gaining by performing analysis. To bypass these issues, the following article introduces a novel framework, known as the Ladon Hybrid Cloud, for preserving cloud-based firewall policy confidentiality. It is shown that in this framework, a high level of privacy is provided thanks to leveraging an anonymized firewall approach and a hybrid cloud model. A number of optimization techniques, which help to further improve the Ladon Hybrid Cloud privacy level, are also introduced. Finally, analysis performed on the framework shows that it is possible to find a trade-off between the Ladon Hybrid Cloud privacy level, its congestion probability, and efficiency. This argument has been demonstrated through the results of conducted experiments.

**Keywords** Firewall · Cloud computing · Privacy · Bloom Filter

✉ Tytus Kurek  
kurek@kt.agh.edu.pl

Marcin Niemiec  
niemiec@kt.agh.edu.pl

Artur Lason  
lason@kt.agh.edu.pl

<sup>1</sup> AGH University of Science and Technology, Mickiewicza 30, 30-059 Krakow, Poland

## 1 Introduction

During the past couple of years, the cloud computing paradigm has evolved from an experimental approach to hosting *Information and Communications Technology* (ICT) services in a distributed systems environment, to a leading trend in the ICT market [1]. Thanks to this, most types of services are available in a cloud today, including security services. The model of hosting security services in a cloud is referred to as *Security as a Service* (SecaaS) [2].

Following the needs of business which keep increasing due to the expansion of the technology, many ICT companies, including leaders such as AT&T with its *Network-Based FireWall Services* (NBFWS) [3], have already begun offering security services in a cloud. These include firewall services, *Intrusion Prevention System* (IPS) services, e-mail filtering, and web filtering. In most cases, including AT&T NBFWS and Cloudera *Enterprise Services Cloud* (ESC) [4], the security services are deployed by leveraging a hybrid cloud model with customers connected to the *Cloud Service Provider* (CSP) via a secure *Virtual Private Network* (VPN) connection. In such a system, most of the customer security services are hosted in a cloud, while the basic security infrastructure, responding to last mile attacks for example, remains on its premises. The on-premises infrastructure can be managed by the CSP or the customer. Alternatively, a hybrid management system can be applied with the CSP being responsible for the on-premises infrastructure installation, its initial configuration, monitoring, etc., and the customer being responsible for the entire security policy management.

One of the core security services adopted by the vast majority of organizations are firewall services. It is hard to imagine an enterprise, government unit, university, or even home business running its network services without being protected by a firewall. Thanks to such technologies as AT&T

NBFWs or Virtela ESC, these can be outsourced to the cloud, resulting in significantly reduced management overhead, decreased *Total Cost of Ownership* (TCO), improved business agility, and so on [5]. However, because of a lack of mechanisms preventing the CSP from having an insight into the customer's firewall policy, there are still issues of information confidentiality and privacy [6–9].

In addition, another threat is information gaining by traffic eavesdropping and analysis. Since in a hybrid cloud SecaaS model all the traffic flows unencrypted through the CSP infrastructure and there are no mechanisms protecting against eavesdropping, sensitive information such as that regarding allowed *Internet Protocol* (IP) addresses can be easily gained by the CSP based on traffic analysis. This exposes a serious vulnerability of such systems, as according to recent reports, most data harvesting events take place during transit [10, 11]. In addition, although the CSP itself is obliged by contract to maintain information confidentiality, according to research shown in [12, 13], employees would not hesitate to steal such sensitive information if laid off, for example.

This leads to the following conclusion. Until mechanisms preserving firewall policy confidentiality and preventing information gaining by traffic eavesdropping and analysis are designed, organizations will not be able to run their firewall services in a cloud in a way that is sufficiently confidential to preserve their privacy. This problem seems to be an unresolved security hole, as only one solution has been proposed so far.

Referred to as the Ladon framework by its authors, it attempts to preserve cloud-based firewall policy confidentiality [14]. It is supposed to achieve it by leveraging an anonymized firewall in the public cloud. In such frameworks, the CSP is prevented from having an insight into the original firewall policy. However, the Ladon framework provides no mechanism for preventing the CSP from deducing the original firewall policy by traffic eavesdropping and analysis. As the final decision on network packets is still known to the CSP, it can determine the original firewall policy over time. As such, the privacy of cloud-based firewall policies cannot be preserved using Ladon.

Motivated by the above observations, the following contributions are made in this paper. Firstly, a novel framework for preserving cloud-based firewall policy confidentiality, known as the Ladon Hybrid Cloud, is introduced as an extension and augmentation to the regular Ladon framework. It is shown that by introducing the purposefulness of packet decision uncertainty, the main drawback of Ladon—the risk of firewall deanonymization by packets eavesdropping and analysis—is significantly reduced. Additional optimization techniques which help improve Ladon Hybrid Cloud privacy level based on the type of firewall policy in use are also introduced. It is shown that after deploying the Ladon Hybrid Cloud according to best practices, the risk of infor-

mation gaining by the CSP does not differ significantly from that of a regular *Internet Service Provider* (ISP). Finally, by performing mathematical framework analysis, the results of which have been confirmed through the results of the experiment performed, the article shows that it is possible to find a trade-off between the Ladon Hybrid Cloud privacy level, its congestion probability, and efficiency.

The rest of the paper proceeds as follows. First of all, related work is reviewed in Sect. 2. Section 3 includes a presentation of the Ladon framework. In Sect. 4, a novel framework for preserving cloud-based firewall policy confidentiality, known as the Ladon Hybrid Cloud, is introduced, along with its optimization techniques. In Sect. 5, all the mathematical framework analyses are shown. Their experimental results follow in Sect. 6. All observed Ladon Hybrid Cloud limitations and directions for future work are noted in Sect. 7. Finally, Sect. 8 contains the conclusions.

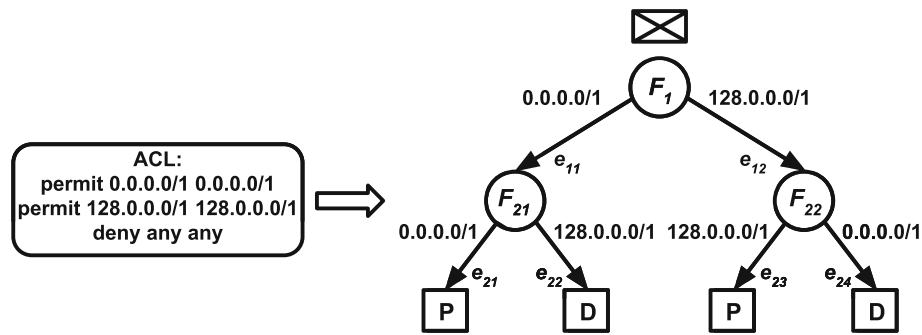
## 2 Related work

Khakpour and Liu [14] presented the Ladon framework as a first step toward cloud-based firewalling. The Ladon leverages an anonymized firewall based on a set of *Bloom Filter Firewall Decision Diagrams* (BFFDDs) which are compiled from regular *Firewall Decision Diagrams* (FDDs) [15] in which edge sets are replaced by *Bloom Filters* (BFs) [16]. Thanks to the merging of these elements that are explained in detail in the next section, regular *Access Control List* (ACL) rules are transformed into a structure which is still visible to the CSP, although it does not provide it with straightforward information regarding the original ACL structure. In such a framework, the ACL rules of the customer's firewall can neither be directly read by the CSP, nor easily cracked using brute-force techniques. However, as described below, these can be determined by packets eavesdropping and analysis.

Other studies related to the topic of this article are those related to moving target defense. In [17], the authors have studied techniques of substituting different targets for any given request in order to create a dynamic and uncertain attack surface area of a given system. This enabled them to demonstrate that such systems are less vulnerable and more secure. The Ladon Hybrid Cloud framework presented in this article also intentionally introduces uncertainty to the attack surface area; however, it achieves this by using a BF false-positive rate, as explained below. All targets remain unchanged for all given requests over time.

## 3 From ACL to Ladon

An FDD, presented by Gouda and Liu in [16], is a mathematical structure which is a formal firewall representation. In



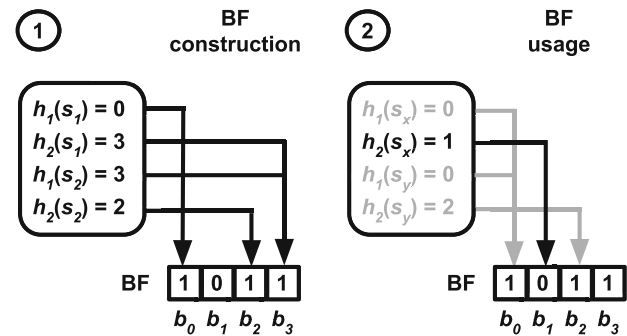
**Fig. 1** FDD construction

fact, the FDD transforms a regular firewall policy based on a set of *Access Control Entry* (ACEs) into a tree where packets pass from top to bottom, with particular packet fields being examined at each level. Depending on its particular packet field value, the packet is directed to one of the edges, forming a decision path which finally takes one of the two possible decisions: permit or deny.

This concept is shown in Fig. 1. Suppose that the firewall takes its final decision based on the source and destination IP addresses alone. The FDD then consists of two levels: one representing the source IP address and the other representing the destination IP address. The edge sets are calculated based on the corresponding ACL. For example, for a packet sourced at 10.10.10.10 and destined for 192.168.192.168, which fits the first ACE in the ACL, its source IP address is examined first on the  $F_1$  node. 10.10.10.10 fits the 0.0.0.0/1 set, so the packet is passed to the  $e_{11}$  edge, where its destination IP address is examined on the  $F_{21}$  node. Because 192.168.192.168 fits the 128.0.0.0/1 set, the packet is passed to the  $e_{22}$  edge, resulting in a deny decision.

Unlike in a regular firewall, where a packet is examined as a whole by testing it against ACEs from top to bottom until the first match is found, the FDD takes a completely different approach. It splits the packet into fields and examines each field independently on particular tree levels. The resulting path leads to a single, ultimate decision. Sample FDD implementation known as ‘Policy Trie’ was also presented independently of Gouda and Liu’s work by Fulp and Tarsa in [18].

A BF, presented by Bloom in [16], is a mathematical probabilistic data structure which is used to test whether an object is a member of a set in a time-efficient manner. Mathematically, a BF is a bit array with a size of  $m$  which is generated by calculating  $k$ -independent hash functions for each of  $n$  elements of the set. For each of the results, the corresponding index in BF is set to 1. To check whether an element is a member of the original set, the same hash functions are calculated and corresponding indexes of the BF are checked. If at least one of them is 0, the element is not a member of the original set. If all of them are 1, the element may be a member of the original set. The above indicates that a BF may



**Fig. 2** BF construction and usage

result in false positives. Moreover, the value of false-positive probability, also known as the BF false-positive rate, can be calculated based on the  $k$ ,  $m$ , and  $n$  parameter [19].

This is shown in Fig. 2. Suppose that a BF with a size of  $m = 4$  using  $k = 2$  hash functions ( $h_1, h_2$ ) represents a set containing  $n = 2$  elements ( $s_1, s_2$ ). The BF is generated (case 1) by calculating hash functions for each of the elements ( $h_1(s_1), h_2(s_1), h_1(s_2), h_2(s_2)$ ) and setting up corresponding indexes in the BF ( $b_0, b_1, b_2$  or  $b_3$ ) to 1. Suppose that the hash function results are as follows:  $h_1(s_1) = 0, h_2(s_1) = 3, h_1(s_2) = 3, h_2(s_2) = 2$ . The following BF indexes are set to 1 as a result:  $b_0, b_2, b_3$ . At this point, the BF can be used to test object presence in the source set (case 2). For a given object ( $s_x$ ) to be tested, the same hash functions are calculated ( $h_1(s_x), h_2(s_x)$ ) and corresponding BF indexes are examined to see whether they are set to 1. Suppose that the hash function results are as follows:  $h_1(s_x) = 0, h_2(s_x) = 1$ . It is then clear that the object is not an element of the source set— $b_1$  is not set to 1. On the other hand, if the hash function results for some other object ( $s_y$ ) to be tested are as follows:  $h_1(s_y) = 0, h_2(s_y) = 2$ , the object is considered to be a member of the source set with some permissible misclassification probability, known as the BF false-positive rate. While the object  $s_y$  is considered to be a member of the source set, it actually is not.

So far, brick level structures which build up BFFDD have been covered. Based on them, the BFFDD definition can be explained as follows. According to [14], the BFFDD is a data

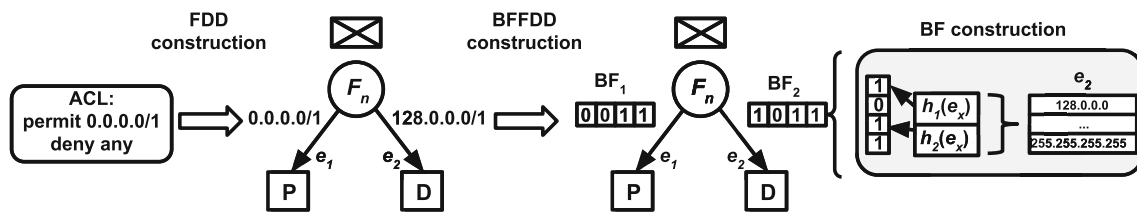


Fig. 3 BFFDD construction

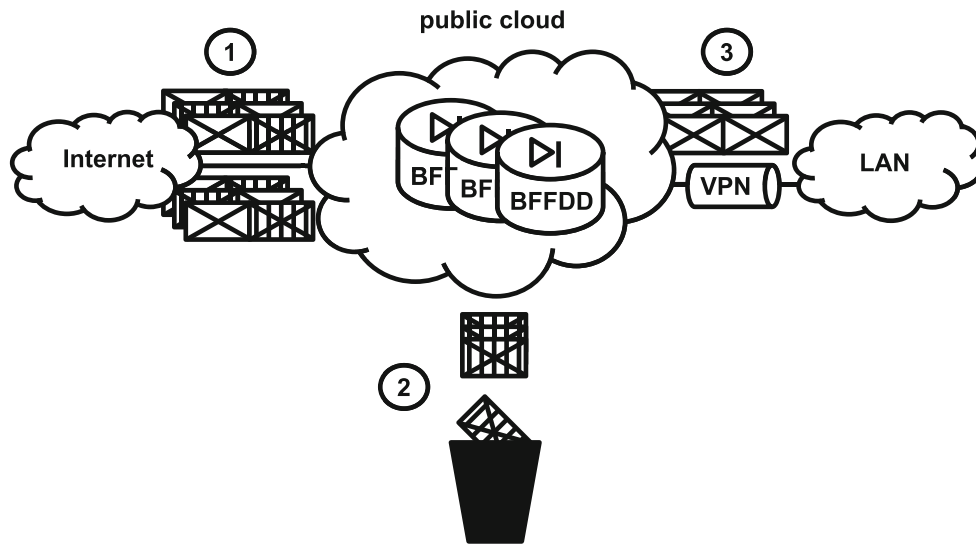


Fig. 4 Ladon framework

structure formed from regular FDD where, for a given edge, the edge set is represented by a BF. Because it is in the nature of the BF that it may result in false positives, ambiguities may occur in BFFDD, leading to multiple decision paths and as a result to multiple decisions. To eliminate such ambiguities  $N$ , independent BFFDDs are implemented and executed simultaneously. The resulting decision paths are then compared looking for a single, common path which leads to a common, final decision.

The concept of BFFDD and its construction algorithm is shown in Fig. 3. Suppose that the firewall takes its final decision based on the source IP address only. The original ACL is then transformed into an FDD with one level only. Next, the edge sets  $e_1(0.0.0.0/1)$  and  $e_2(128.0.0.0/1)$  are transformed into  $BF_1$  and  $BF_2$  correspondingly. The process of  $BF_2$  construction is shown within the gray round rectangle. However, the BF shown in this example has a size of  $m = 4$  and uses  $k = 2$  hash functions; these are obviously much greater in a real scenario.

Section 1 shows a standard business model for hosting firewall services in the cloud. Outsourced firewall services are hosted in the public cloud located in the data center owned and managed by the CSP. All traffic destined to the customer first enters the public cloud, which is connected with customer premises via a secure VPN connection. The tech-

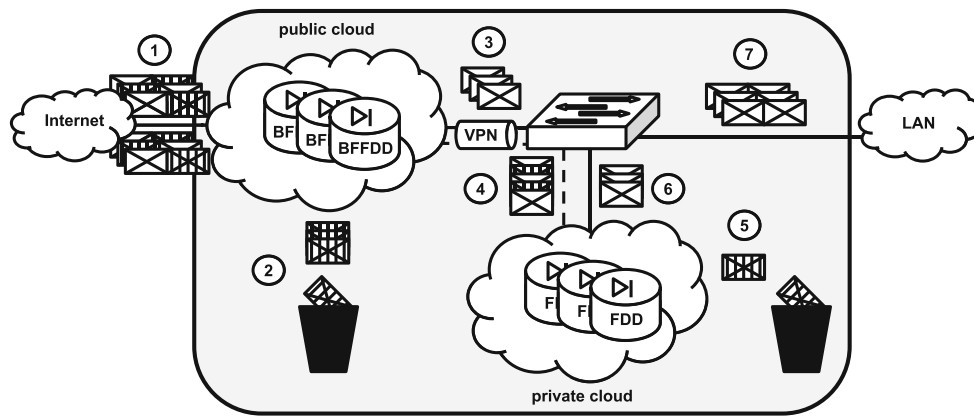
nology used to deliver firewall services is not visible to the customer. Assume that it is based on a set of independent BFFDDs as described above. A framework of cloud-based firewall services based on BFFDDs is shown in Fig. 4.

Packets permitted on customer premises, referred to as 'good packets' in the rest of the article, are represented there by plain envelopes. In turn, packets denied on customer premises, referred to as 'bad packets' in the rest of the article, are represented there by striped envelopes. All packets enter the public cloud first (step 1) where bad packets are discarded (step 2). Next, good packets are sent to customer LAN (step 3). Such framework was referred to as the Ladon framework by its authors in [14].

By implementing and testing Ladon in a live environment, Khakpour and Liu demonstrated that it is an effective framework for the outsourcing of the firewall services. It was also shown that any attempts to deanonymize the BFFDD can be extremely time-consuming.

## 4 Framework design

Assume there is Ladon framework implemented with the firewall services hosted by an honest-but-curious CSP wishing to get an insight into the customer's firewall security pol-



**Fig. 5** Ladon hybrid cloud

icy. The CSP has an insight into the BFFDDs delivering the firewall services, but as these are just binary structures representing the BFs on the edges of the original FDDs used to generate the BFFDDs, it gives it no information on the original firewall security policy. However, the CSP has an insight into whether a packet flowing via the Ladon framework is permitted on the customer side. This is because only packets permitted on the customer side are flowing via the VPN connection between the CSP and the customer. Packets denied on the customer side are discarded in the public cloud. Although the VPN connection is encrypted, the CSP, as one of its initiators, can intercept packets entering the VPN.

Therefore, although the regular Ladon framework resolves the issue of firewall policy confidentiality in such a way that it cannot be directly read, the final decision for a packet is still known by the CSP. This means that after an appropriately long period of time, the CSP can build up an almost full knowledge base regarding packets which are permitted on the customer side. While the underlying ACL structure is protected, traffic flowing between the CSP and the customer can still be easily eavesdropped and analyzed to obtain information about good packets.

Such a framework allows the CSP to bypass the reconnaissance phase, and most of the scanning phase of an attack, which according to [20], can significantly reduce the time required to perform the attack. To address this issue, a novel framework, called the Ladon Hybrid Cloud, which introduces the purposefulness of packet decision uncertainty based on BFFDD and a hybrid cloud model, was designed and is presented in the following section of the article.

#### 4.1 Introducing the Ladon hybrid cloud

In a regular Ladon framework, based on a set of BFFDDs, the final decision for a packet is always certain. However, in a single BFFDD, ambiguities may occur as a result of BF false positives. This is because a BF false positive leads to

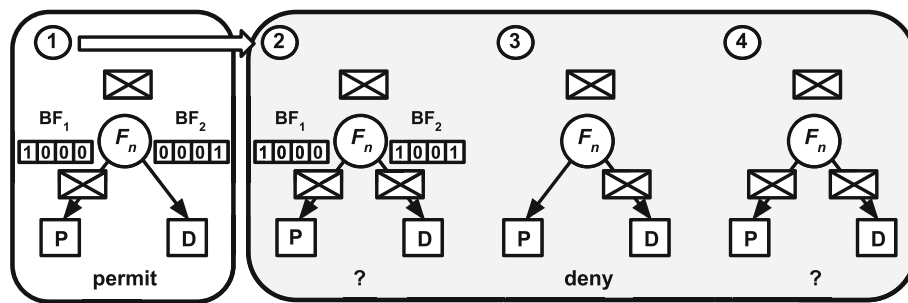
a situation where the packet field matches both edges on a particular BFFDD level, resulting in multiple decision paths, and as a result in multiple decisions for the packet.

Instead of eliminating such ambiguities by using a set of BFFDDs and searching for a single, common path with a single, common decision, the Ladon Hybrid Cloud takes a completely different approach. It leverages a single BFFDD instead and intentionally allows some of the packets to result in multiple decisions. This approach leverages a hybrid cloud model, as it is a leading trend in the SecaaS market [3,4], with BFFDD in a public cloud and a regular firewall in a private cloud. In order to simplify the management of the private cloud, hybrid management can be applied.

In this novel framework, after passing through BFFDD in a public cloud on the CSP side, packets resulting in certain deny decisions are directly discarded, while those resulting in certain permit decisions are sent directly to the customer over a trusted network. Additionally, packets resulting in multiple decisions are sent to the private cloud on customer premises over an untrusted network for additional filtering. Segregation of packets resulting in certain permit decisions from those resulting in multiple decisions can be organized based on the *Virtual Local Area Network* (VLAN) logic.

The Ladon Hybrid Cloud concept is shown in Fig. 5 inside the gray round rectangle. Good packets are represented there by plain envelopes, while bad packets are represented by striped envelopes. All packets enter the public cloud first (step 1) where those resulting in certain deny decisions are directly discarded (step 2). Those resulting in certain permit decisions are sent directly to the customer *Local Area Network* (LAN) over a trusted VLAN, represented by a continuous line (steps 3 and 7). Finally, packets resulting in multiple decisions are sent to the private cloud for additional filtering over an untrusted VLAN, represented by the dotted line (step 4). The private cloud performs additional filtering by discarding of the rest of the denied packets (step 5) and sends permitted packets into the customer LAN (step 6).





**Fig. 6** BFFDDCF construction

Although customer traffic is still exposed to eavesdropping and analysis by the CSP in the Ladon Hybrid Cloud, the amount of information carried by particular packets is reduced compared to a regular Ladon framework. This is because some packets result in multiple decisions after passing through BFFDD which causes the knowledge base built by the hostile CSP not to be 100% accurate. Moreover, assuming that over time the number of bad packets with different packet headers is growing, packets diversity causes the knowledge base to grow too. The above assumption takes into account *Distributed Denial of Service* (DDoS) attacks, for example, during which most of the source IP addresses are new [21].

An inaccurate and constantly growing knowledge base regarding packets perceived by the CSP as good packets causes the CSP not to fully trust it and forces it to perform additional reconnaissance and scanning in order to obtain accurate information. Moreover, over time, demand for resources required to store and analyze the data can significantly increase. It is therefore clear that the Ladon Hybrid Cloud eliminates the main drawback of a regular Ladon framework: the risk of firewall deanonymization by packets eavesdropping and analysis. This is achieved by introducing the purposefulness of packet decision uncertainty. The following sections cover additional framework optimization mechanisms which help increase this uncertainty even further and as a result provide a high level of privacy.

## 4.2 Ladon hybrid cloud optimization

Although the BFFDD may result in multiple decisions for some of the packets, for the others, a final decision remains certain. Part of the knowledge base maintained by the hostile CSP will therefore always be accurate. It is possible to eliminate this vulnerability, however, by redesigning the BFFDD in such a way that it always results in multiple decisions for either good or bad packets, based on the adopted firewall policy type.

In the real world, two types of firewall policies can be adopted based on organization requirements:

- Closed: Permitting only a specific subset of traffic and denying the rest,
- Open: Denying only a specific subset of traffic and permitting the rest.

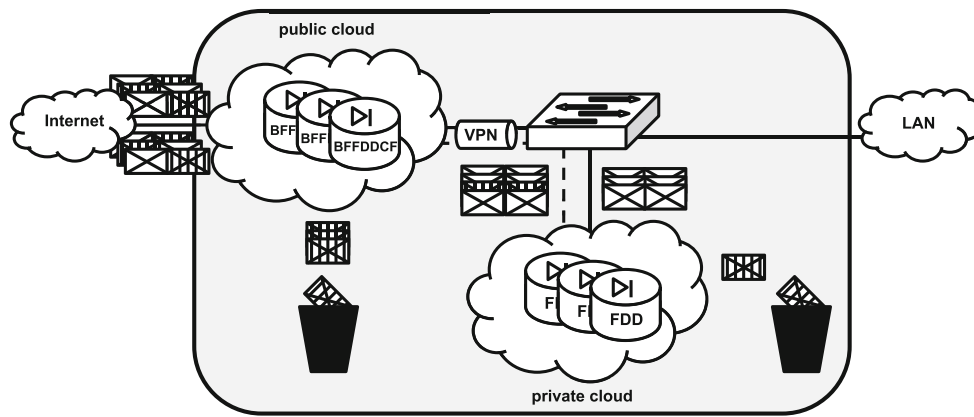
For inbound traffic flow, considered in this article, most organizations apply the closed firewall policy rather than the open one, because it minimizes the risk of malicious traffic passing through. In such a case, the BFFDD is redesigned in such a way that it always results in multiple decisions for good packets. As closed firewall policy is a leading trend in most of the organizations today, it will be used as an example in further arguments in this article. Likewise, in an organization applying open firewall policy, the BFFDD can be redesigned so it always results in multiple decisions for bad packets accordingly.

As has been mentioned, in the case of a closed firewall policy type, the BFFDD is updated to always result in multiple decisions for good packets. The only packets that may still result in certain decisions are therefore bad packets. The framework is designed in this way, because when adopting closed firewall policy type, good packets carry significantly more information for the CSP regarding the original ACL structure compared to bad packets. This is because a characteristic of closed firewall policy type is that the subset of traffic which is permitted is much smaller than the subset of traffic which is denied.

Figure 6 represents a BFFDD with one level and all the cases that it can result in:

- Case 1: Certain permit decision for good packets,
- Case 2: Multiple decisions for good packets,
- Case 3: Certain deny decision for bad packets,
- Case 4: Multiple decisions for bad packets.

As mentioned above, case 1 should be fully eliminated. To achieve this, a regular BFFDD is compiled first and then tested against all good packets. For those resulting in certain permit decisions (case 1), the BF representing the set of the edge that leads to a deny decision (BF<sub>2</sub> in this case) is updated



**Fig. 7** Ladon hybrid cloud for closed firewalls

so that it results in a forced false positive. As a consequence, multiple decision paths are applied to all good packets (case 2) leading to multiple decisions applied to all of them. In other words, the redesigned BFFDD eliminates case 1 by transforming it into case 2, resulting in three possible cases (2, 3, and 4) shown inside the gray round rectangle.

The above transformation can be performed on any BFFDD level; however, for analysis and implementation simplicity, it is assumed that it is performed on the last level representing the last examined packet field. Such a redesigned BFFDD will be referred to as BFFDDCF (*Bloom Filter Firewall Decision Diagram for Closed Firewalls*). Likewise, such a redesigned Ladon Hybrid Cloud which leverages a BFFDDCF in a public cloud will be referred to as *Ladon Hybrid Cloud for Closed Firewalls* (LHCCF).

As a consequence, the traffic flowing between public and private clouds consists of all good packets and some bad packets, while multiple decisions are applied to all of them by the BFFDDCF. This leads to a situation where all packets flowing between the CSP and the customer go via an untrusted VLAN represented by the dotted line, so the traffic segregation engine can be fully eliminated from the LHCCF, as shown in Fig. 7. As all good packets require additional filtering in the private cloud, there is no traffic flowing between public and private clouds over the trusted VLAN represented by the continuous line. It is clear that in this case, the private cloud needs to process more packets.

In such a framework, the CSP cannot draw any additional information from the traffic, except of a fact that part of it is permitted on customer premises. However, the CSP can still maintain a certain knowledge base regarding packets which are explicitly denied in the BFFDDCF, and this gives it a very limited amount of information, as has been stated before. At this point, the amount of information which the CSP can extract by performing traffic eavesdropping and analysis does not differ greatly from that of the regular ISP which the customer is connected to. The original firewall policy cannot be directly read by the CSP or be assumed by performing traffic

eavesdropping and analysis. While the first of these two features is provided by the regular Ladon framework, the second is provided by the Ladon Hybrid Cloud only.

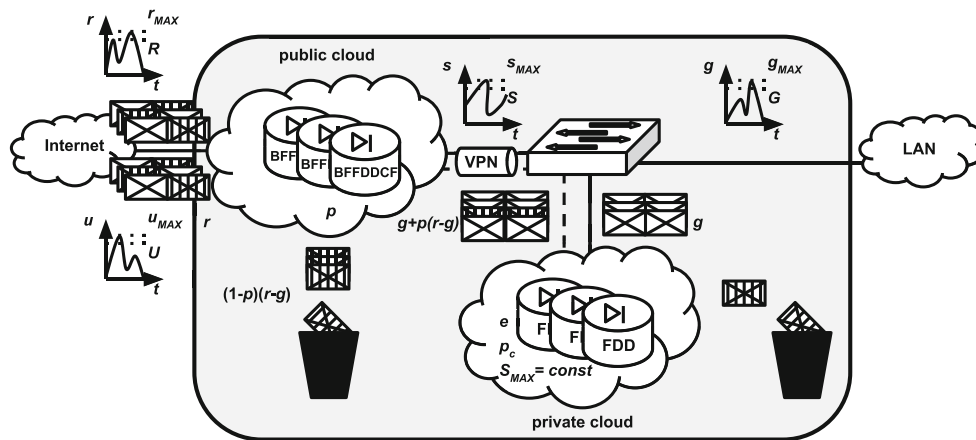
## 5 Framework analysis

Because of the uncertainty of packet decision making in the LHCCF, the rate of traffic flowing between public and private clouds is increased compared to a regular Ladon framework. This is because the traffic consists not only of good packets, but also of some bad packets which result in multiple decisions after passing through the BFFDDCF. The factor of bad packets which are transmitted is determined by the probability that the BFFDDCF results in multiple decisions, referred to as 'BFFDDCF multiple decision probability' in the rest of this article. Furthermore, the BFFDDCF multiple decision probability is a result of the false-positive rates of the particular BFs that make it up. The following section shows how the parameters of the particular BFs in the BFFDDCF can be used to control the rate of bad packets flowing between public and private parts of the LHCCF. As a result, an overall rate of traffic flowing between the CSP and the customer can be controlled as well and a trade-off can be found between the LHCCF privacy level, its congestion probability, and efficiency.

### 5.1 Controlling traffic rate

So far, it has been shown that the greater the value of the BFFDDCF multiple decision probability is, the higher the privacy level is provided by the LHCCF. This is because increasing the BFFDDCF multiple decision probability increases the number of bad packets that result in multiple decisions and, as a result, decreases the amount of information carried by particular packets.

But can the BFFDDCF multiple decision probability be increased without limits? What is its effect on LHCCF con-



**Fig. 8** Traffic flow in the LHCCF

gestion probability and efficiency? The following analysis attempts to answer these questions. Let us define:

- $p$ —BFFDDCF multiple decision probability
- $r$ —Rate of traffic at the public cloud entrance (packets/s)
- $s$ —Rate of traffic at the public cloud exit (packets/s)
- $g$ —Rate of traffic consisting of good packets (packets/s)
- $u$ —Good packet ratio
- $S_{MAX}$ —Throughput of the private cloud
- $p_c$ —LHCCF congestion probability
- $e$ —LHCCF efficiency

These are shown in Fig. 8.

Based on the above, the rate of traffic at the public cloud exit is:

$$s = g + p(r - g) \quad (1)$$

as it consists of all good packets and those bad packets that result in multiple decisions after passing through the BFFDDCF. It is then possible to control the rate of traffic at the public cloud exit while knowing the value of the BFFDDCF multiple decision probability, the rate of traffic at the public cloud entrance, and the rate of traffic consisting of good packets.

By expressing the ratio, referred to as ‘good packet ratio’ in the rest of the article, of the rate of traffic consisting of good packets and the rate of traffic at the public cloud entrance as:

$$u = \frac{g}{r} \quad (2)$$

Formula 1 can be transformed to a function of good packet ratio and is:

$$s = ur + p(r - ur) \quad (3)$$

Again, it is then possible to control the rate of traffic at the public cloud exit knowing the value of the BFFDDCF multiple decision probability, the rate of traffic at the public cloud entrance, and the good packet ratio.

The rate of traffic at the public cloud exit is also the rate of traffic at the private cloud entrance. It is therefore important to control the rate of traffic at the private cloud entrance to ensure that it never exceeds its throughput. This is because a higher traffic rate results in private cloud congestion and, as a result, in congestion of the whole LHCCF. The equation from Formula 3 can be then replaced with the inequality:

$$S_{MAX} \geq ur + p(r - ur) \quad (4)$$

Therefore, by knowing the values of the rate of traffic at the public cloud entrance and good packet ratio, it is then possible to adjust the BFFDDCF multiple decision probability value so that the rate of traffic at the private cloud entrance never exceeds its throughput.

The values of the rate of traffic at the public cloud entrance and good packet ratio change over time; however, the BFFDDCF multiple decision probability cannot change over time, because of computational limitations as discussed below. Therefore, while designing BFFDDCF, some constant values of the rate of traffic at the public cloud entrance ( $R$ ) and the good packet ratio ( $U$ ) need to be assumed instead.

Those could be taken based on maximum values of particular variables ( $r_{MAX}$ ,  $u_{MAX}$ ) observed over time to ensure LHCCF congestion probability on the lowest possible level. However, as shown later in this section, high values of  $R$  and  $U$  enforce a lower BFFDDCF multiple decision probability and then, as a result, a lower LHCCF privacy level. As such, instead of selecting maximum values, it is better to relax them somewhat (e.g., by selecting average values). On the other hand, lower  $R$  and  $U$  values result in a higher privacy level while allowing LHCCF congestion during  $r$  and  $u$  peak periods. This is because instantaneous  $r$  and  $u$  values exceeding



$R$  and  $U$  constants result in the rate of traffic at the private cloud entrance exceeding its throughput.

Another parameter to consider when designing the LHCCF is a time period for which constant values of  $R$  and  $U$  are selected. As traffic statistics tend to be similar over recurrent time periods, it may be worth running different BFFDDCFs during these periods (e.g., one BFFDDCF during the day and another during the night). Thus, constant values of  $R$  and  $U$  should be selected, and the following analysis should be performed for each recurrent time period independently when designing the LHCCF.

Assuming that statistics of the rate of traffic at the public cloud entrance are known, it is then possible to select a constant value:

$$R \approx r_{\text{MAX}} \quad (5)$$

while ensuring that the instantaneous value of  $r$  variable exceeds  $R$  constant with a probability of:

$$p_r = P(r > R) \quad (6)$$

Likewise, assuming that good packet ratio statistics are also known, it is then possible to select a constant value:

$$U \approx u_{\text{MAX}} \quad (7)$$

while ensuring that the instantaneous value of the  $u$  variable exceeds the  $U$  constant with a probability of:

$$p_u = P(u > U) \quad (8)$$

Based on approximations made in Formulas 6 and 8, the LHCCF congestion probability is as follows:

$$\begin{aligned} p_c &= P(r > R \cup u > U) \\ &= p_r + p_u - P(r > R \cap u > U) \end{aligned} \quad (9)$$

It can also be expressed as follows:

$$p_c = P(s > S_{\text{MAX}}) = P((ur + p(r - ur)) \geq S_{\text{MAX}}) \quad (10)$$

Assuming that both fragmentary probabilities and the probability of a conjunction in Formula 9 are known based on the traffic analysis, it is then possible to control the LHCCF congestion probability.

Moreover, it can be seen that the higher the value of the BFFDDCF multiple decision probability, the higher the LHCCF congestion probability. On the other hand, it has already been shown that the higher the value of the BFFDDCF multiple decision probability, the higher the LHCCF privacy level. Thus, finding a trade-off between the LHCCF privacy level and its congestion probability is a matter of

selecting such a value of the BFFDDCF multiple decision probability that satisfies both.

Let us also define the LHCCF efficiency as a ratio of the rate of the traffic at the public cloud entrance and the rate of the traffic at the public cloud exit. It is then:

$$e = \frac{r}{s} = \frac{1}{u + p(1 - u)} \quad (11)$$

However, as the  $R$  constant value has been assumed instead, the LHCCF efficiency is constant too and is:

$$E = \frac{1}{U + p(1 - U)} \quad (12)$$

It is then possible to control the LHCCF efficiency while knowing the value of the BFFDDCF multiple decision probability.

Moreover, it can be seen that the higher the value of the BFFDDCF multiple decision probability, the lower the LHCCF efficiency. Thus, finding a trade-off between the LHCCF privacy level and its efficiency is a matter of selecting such a value of the BFFDDCF multiple decision probability that satisfies both. Finally, finding a trade-off between the LHCCF privacy level, its congestion probability and efficiency is a matter of selecting such a value of the BFFDDCF multiple decision probability that satisfies all three parameters.

By replacing the  $r$  and  $g$  variables in Formula 4 by the  $R$  and  $U$  constants from Formulas 5 and 7, it is transformed thus:

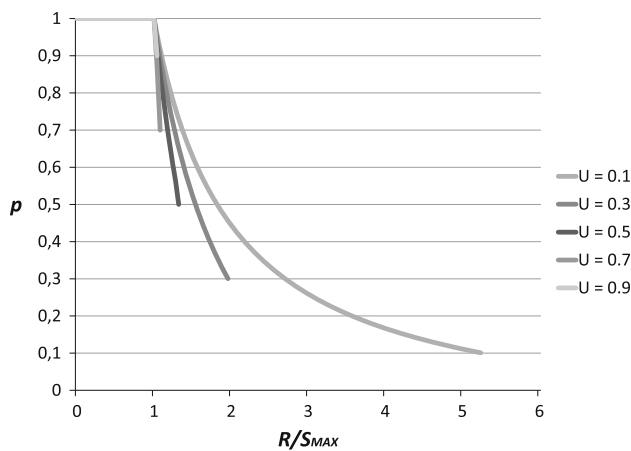
$$S_{\text{MAX}} \geq UR + p(R - UR) \quad (13)$$

therefore, the BFFDDCF multiple decision probability is:

$$p \leq \frac{S_{\text{MAX}} - UR}{R(1 - U)} \quad (14)$$

It is then possible to control the rate of traffic at the private cloud entrance by adjusting the BFFDDCF multiple decision probability value while knowing private cloud throughput and base statistical traffic parameters. These include the rate of traffic at the private cloud entrance and the good packet ratio, both on an agreed level.

As shown in the next subsection, the BFFDDCF multiple decision probability can take any value between  $U$  and 1. By taking advantage of this property ( $p \in (U, 1]$ ) and by substituting edge values into Formula 10, it can be seen that it applies to the  $R \in [S_{\text{MAX}}; \frac{S_{\text{MAX}}}{U(2-U)}]$  interval only. This is because for the rate of traffic at the public cloud entrance being lower than the private cloud throughput, the BFFDDCF multiple decision probability can take any value from the  $(U; 1]$  interval. The generalized Formula 14 is then:



**Fig. 9** Maximum allowable BFFDDCF multiple decision probability

$$p \leq \begin{cases} 1 & \text{for } R \in [0; S_{MAX}) \\ \frac{S_{MAX}-UR}{R(1-U)} & \text{for } R \in [S_{MAX}; \frac{S_{MAX}}{U(2-U)}) \end{cases} \quad (15)$$

A plot of the maximum allowable BFFDDCF multiple decision probability of the  $R$  argument including the  $U$  parameter is shown in Fig. 9.

## 5.2 BF false-positive rate: the core control engine

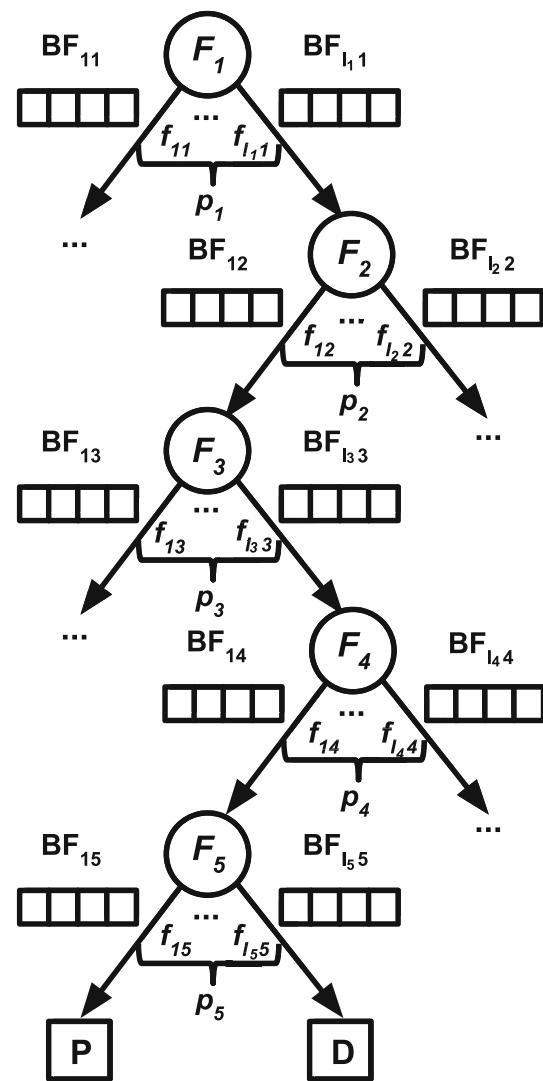
So far, it has been shown that the BFFDDCF multiple decision probability can be used to control the rate of traffic flowing between the CSP and the customer. The following section takes a closer look at the BFFDDCF multiple decision probability itself, which is the result of false-positive rates of particular BFs which make it up.

Based on studies conducted in [18], a classic FDD consists of five fields which include protocol, source IP, source port, destination IP, and destination port. Suppose that the BFFDDCF presented in the following example consists of the same five fields and has  $j = 5$  levels as shown in Fig. 10. Let us also define:

- $p_g$ —BFFDDCF multiple decision probability for good packets,
- $p_b$ —BFFDDCF multiple decision probability for bad packets,
- $p_j$ —Comprehensive multiple decision probability on the  $j$ -th BFFDDCF level,
- $f_{ij}$ —False-positive rate of the  $i$ -th BF on the  $j$ -th BFFDDCF level.

These are represented in Fig. 10 as values associated with particular edges and values associated with particular summary buckles, respectively.

The BFFDDCF multiple decision probability is a sum of the BFFDDCF multiple decision probability for good pack-



**Fig. 10** BFFDDCF internal structure

ets multiplied by the good packet ratio and the BFFDDCF multiple decision probability for bad packets multiplied by the bad packet ratio; it is then:

$$p = Up_g + (1 - U)p_b = U + (1 - U)p_b \quad (16)$$

This is because good packets, which make up  $U * 100\%$  of the total traffic, always result in multiple decisions.

The BFFDDCF multiple decision probability for bad packets is a comprehensive multiple decision probability on at least one of its  $j$  levels and is then:

$$p_b = 1 - (1 - p_1)(1 - p_2)(1 - p_3)(1 - p_4)(1 - p_5) \quad (17)$$

In other words, it is a probability of an opposite event that ambiguity occurring on none of  $j$  levels. No ambiguities result in comprehensive multiple decision probabilities on

each BFFDDCF level equal to 0 and in the BFFDDCF multiple decision probability equal to 0 as well. On the other hand, even a single ambiguity on any of  $j$  levels leads to multiple decision paths and to multiple decisions. The above is an internal FDD characteristic [15].

On each of the five levels, multiple decisions for bad packets occur when at least one of the BFs on this level results in a false positive. Furthermore, all BFs on the same level cannot result in a false positive at the same time—at least one of them must result in a true positive. Because of this, the comprehensive multiple decision probability for bad packets on the  $j$ -th level is as follows:

$$p_j = 1 - \prod_{i=1}^{I_j} f_{ij} - \prod_{i=1}^{I_j} (1 - f_{ij}) \quad (18)$$

In other words, it is a probability of an opposite event that none or all of the BFs on the  $j$ -th level result in a false positive.

Formula 18 applies to BFFDDCFs without edges with a set containing all possible values of a particular packet field (known as ‘default edges’ [14]). This is because in the BFFDDCF with default edges on a particular level, ambiguities cannot occur on that level, as the packet field always matches the default edge while it never matches an opposite edge. As a result, comprehensive multiple decision probability on that level is equal to 0, which can then result in the BFFDDCF multiple decision probability being equal to 0 as well. To ensure an absence of default edges in BFFDDCF when designing the ACL used to compile it, it is important to avoid general statements, but to be as strict as possible instead. However, if the above cannot be completed because of the internal ACL structure (e.g., a decision is taken regardless of the destination IP address), some redundant statements can be added which eliminate default edges while not breaking ACL structure. This can be completed by modifying the ACL to work differently for destination IP addresses which do not actually belong to the customer.

By putting Formulas 17 and 18 into Formula 16, the BFFDDCF multiple decision probability is then:

$$p = U + (1 - U) \left( 1 - \prod_{j=1}^5 \left( \prod_{i=1}^{I_j} f_{ij} + \prod_{i=1}^{I_j} (1 - f_{ij}) \right) \right) \quad (19)$$

Now suppose that the presented BFFDDCF consists of  $J$  levels instead of five. By performing analysis as shown above, it is possible to find a generalized version of Formula 19 for the  $J$ -level BFFDDCF multiple decision probability which is:

$$p = U + (1 - U) \left( 1 - \prod_{j=1}^J \left( \prod_{i=1}^{I_j} f_{ij} + \prod_{i=1}^{I_j} (1 - f_{ij}) \right) \right) \quad (20)$$

The above equation cannot be solved by analytical methods. However, knowing that the BF false-positive rate is [16, 19]:

$$f \approx \left( 1 - \left( 1 - \frac{1}{m} \right)^{kn} \right)^k \quad (21)$$

it is possible to design BFFDDCF so that its maximum allowable multiple decision probability is as close to the value from Formula 12 as possible by selecting appropriate values of the BFs that build it up. As  $n$ , which represents the edge set size, is known, these parameters include the number of hash functions ( $k$ ) and BF size ( $m$ ).

Consequently, it is possible to design BFFDDCF by selecting appropriate values of BF parameters so that the rate of traffic at the private cloud entrance can be controlled while knowing the private cloud throughput, base statistical traffic parameters, and the particular edge set sizes. Selecting appropriate BF parameters makes it possible to find a trade-off between Ladon Hybrid Cloud privacy level, its congestion probability, and efficiency.

### 5.3 Summary

The BFFDDCF multiple decision probability affects the rate of traffic at the public cloud exit, both with the rate of traffic at the public cloud entrance and good packet ratio. The rate of traffic at the public cloud exit is also the rate of traffic at the private cloud entrance and is limited by the private cloud throughput. By replacing the rate of traffic at the public cloud entrance and the good packet ratio variables with some constant values, the rate of traffic at the private cloud entrance depends on the BFFDDCF multiple decision probability only. These constant values are calculated based on the traffic parameters while ensuring LHCCF congestion probability and its efficiency at a desired level.

Moreover, the BFFDDCF multiple decision probability depends on false-positive rates of the particular BFs that make it up. BFs can then be designed in such a way that their false-positive rates result in a desired value of the BFFDDCF multiple decision probability. Finally, as the BF false-positive rate depends on particular BF parameters, these can be used as a core engine to control the rate of traffic flowing between the CSP and the customer. Adjusting these parameters and then their derivatives makes it possible to find a trade-off between LHCCF privacy level, its congestion probability, and efficiency.

## 6 Experimental results

In order to confirm the validity of the analysis performed and to demonstrate the veracity of its key findings, the fol-

lowing experiments were conducted. First, a simulation of LHCCF was performed based on traffic statistics from two real firewalls. The experiment demonstrated in practice that it is possible to find a trade-off between the LHCCF privacy level, its congestion probability, and efficiency by selecting an appropriate value of the BFFDDCF multiple decision probability. Next, software for generating BFFDDCF and testing purposes was implemented. It was demonstrated in practice that it is possible to design BFFDDCF with the desired value of multiple decision probability by selecting appropriate parameters of BF's that build it up.

### 6.1 BFFDDCF multiple decision probability selection

Section 5 shows that the core parameters of the LHCCF which affect the LHCCF privacy level, its congestion probability, and efficiency, are the rate of traffic at the public cloud entrance, good packet ratio, and throughput of the private cloud. It has been shown that while throughput of the private cloud is constant, the remaining parameters change over time. As such, the analysis has been based on the constant values of the rate of the traffic at the public cloud entrance and the good packet ratio and selected for each recurring period of time. Contrary to the analysis from Sect. 5, the following simulations demonstrated in practice how the instantaneous values of the LHCCF congestion probability and efficiency change over time depending on real-time statistics of the rate of traffic at the public cloud entrance and the good packet ratio.

The real-time statistics of the rate of the traffic at the public cloud entrance were gathered from two real firewalls, referred to as 'Firewall 1' and 'Firewall 2' in the rest of the section, observed during a one-day period, and sampled every five minutes. These are shown in Fig. 11a, b, respectively. It was assumed that Firewall 1 and Firewall 2 represent the BFFDDCF service hosted in the public cloud of the LHCCF. In turn, the real-time statistics of the good packet ratio were generated separately for Firewall 1 and Firewall 2 using the Chi-squared distribution with an average of 0.7 and 0.3, respectively. The statistics were generated for a one-day period with particular points representing samples taken every five minutes. These are shown in Fig. 11c, d, respectively. A 10Mb/s value of the throughput of the private cloud was assumed for both firewalls.

Next, the following simulations were performed for each firewall. First, the BFFDDCF multiple decision probability with a value of 1 was assumed, and instantaneous values of the LHCCF congestion probability and its efficiency were computed for each sample. Average values of the above parameters were computed later. These are referred to as 'reference values' in the rest of this section. The same method was used to calculate average values of the LHCCF congestion probability and its efficiency for 11 values of the

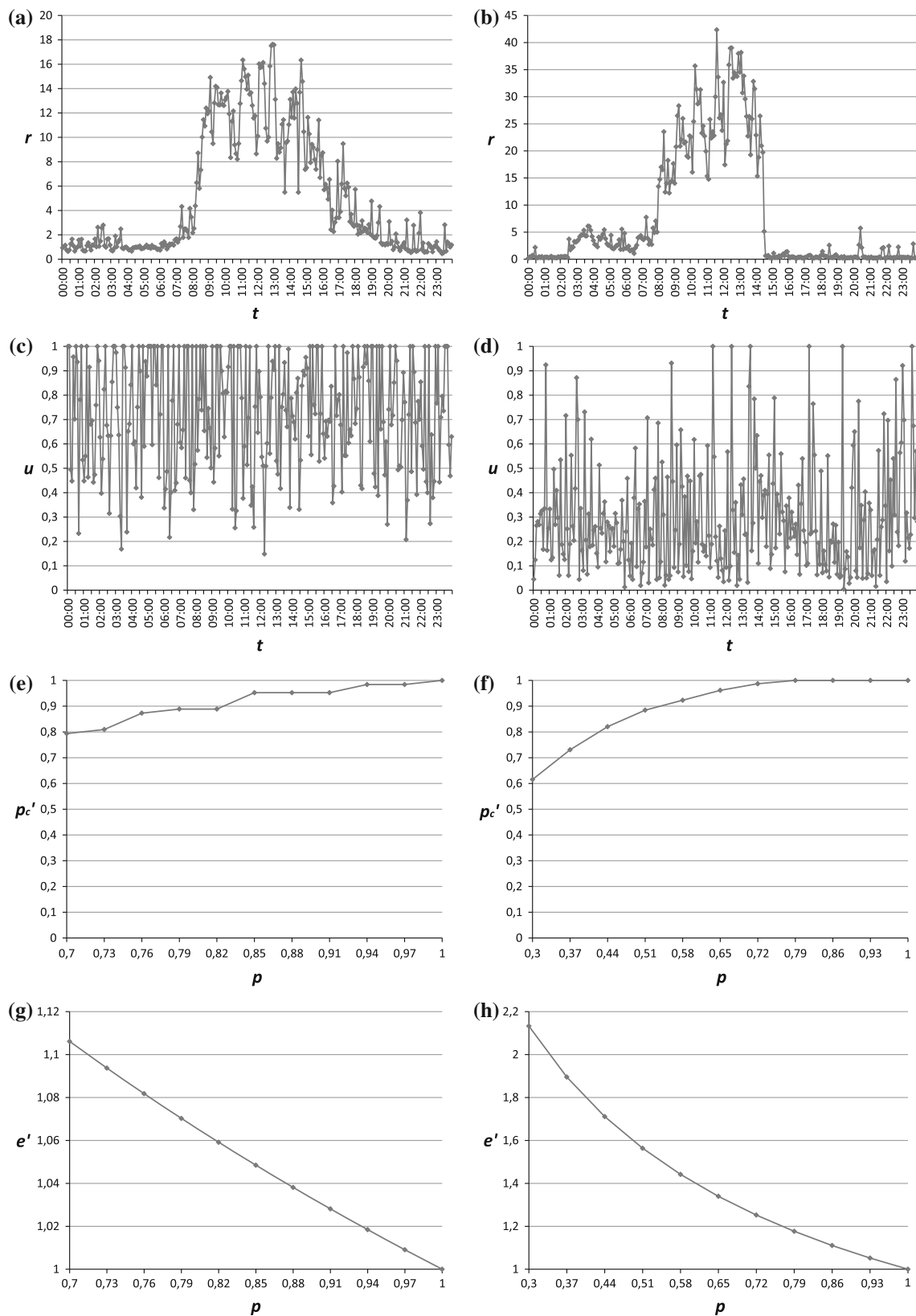
BFFDDCF multiple decision probability, varying from its minimum value of 0.7 and 0.3, respectively, to its maximum value of 1. Finally, rates of the average values and reference values were computed. These are shown in Fig. 11e–h, and they are referred to as 'LHCCF congestion probability ratio' ( $p'_c$ ) and 'LHCCF efficiency ratio' ( $e'$ ), respectively.

It can be seen that the experimental results confirm the validity of conclusions drawn in Sect. 5: the higher the value of the BFFDDCF multiple decision probability, the higher the LHCCF congestion probability and the lower its efficiency. There is no easy way of simulating the LHCCF privacy level, but its relation to the BFFDDCF multiple decision probability is intuitive: the higher the value of the BFFDDCF multiple decision probability, the higher the LHCCF privacy level. Therefore, each of these parameters can be controlled by selecting an appropriate value of the BFFDDCF multiple decision probability. Thus, finding a trade-off between them is possible and is a matter of selecting such a value of the BFFDDCF multiple decision probability that satisfies business requirements. Although there is no single optimal value, a satisfactory range can be found by knowing basic traffic statistics of the designed LHCCF and by defining minimum values of its privacy level and efficiency, and the maximum value of its congestion probability.

### 6.2 BF parameters selection

The second part of the experiment aimed to demonstrate that it is possible to design BFFDDCF with the desired value of multiple decision probability. Thus, a program which generates BFFDDCF on the basis of the set of ACL rules and statistical traffic parameters was implemented and used. This software is referred to as 'generator' in the rest of the article. Another program which computes the BFFDDCF multiple decision probability based on the set of ACL rules and statistical traffic parameters was implemented and used by the authors too. This second piece of software is referred to as 'tester' in the rest of the article. The experiment aimed to use software to compare two values of BFFDDCF multiple decision probabilities: one determined by the generator and the other computed by the tester.

The basic operation of the generator is described below. First, the generator computes FDD based on a given set of ACL rules. Then, it determines the value of the BFFDDCF multiple decision probability based on given statistical traffic parameters, including the throughput of the private cloud, good packet ratio, and rate of traffic at the public cloud entrance, and analysis performed in Sect. 5.1. Knowing the FDD structure and the desired value of the BFFDDCF multiple decision probability, the generator determines BF parameters for each of the BF's in BFFDDCF based on analysis performed in Sect. 5.2. Knowing these parameters, it then transforms FDD to BFFDD by computing the BF's for each



**Fig. 11** Experimental results—BFFDDCF multiple decision probability selection. **a** Firewall 1—rate of the traffic at the public cloud entrance, **b** Firewall 2—rate of the traffic at the public cloud entrance, **c** Firewall 1—good packet ratio, **d** Firewall 2—good packet ratio, **e** Firewall 1—

LHCCF congestion probability ratio, **f** Firewall 2—LHCCF congestion probability ratio, **g** Firewall 1—LHCCF efficiency ratio, **h** Firewall 2—LHCCF efficiency ratio



**Table 1** Experimental results

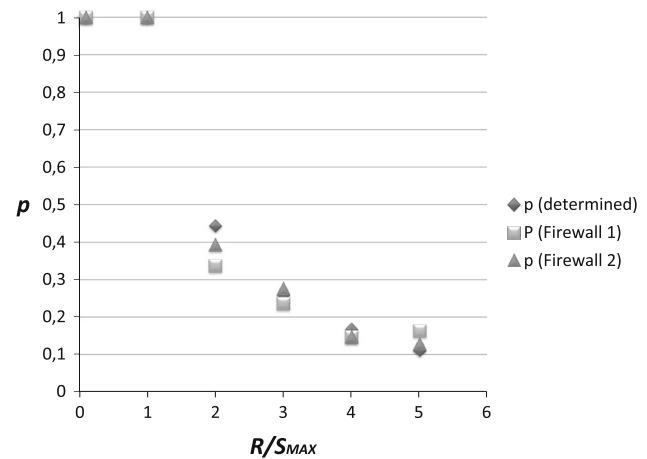
$R$	$p$ (Determined)	$p$ (Firewall 1)	$p$ (Firewall 2)
$0.1S_{MAX}$	1	1	1
$S_{MAX}$	1	1	1
$2S_{MAX}$	0.444	0.337	0.393
$3S_{MAX}$	0.259	0.236	0.276
$4S_{MAX}$	0.167	0.147	0.147
$5S_{MAX}$	0.111	0.163	0.128

of the FDD edges. Finally, the generator transforms BFFDD to BFFDDCF by updating the BF's in such way that good packets always result in multiple decisions.

The multiple decision probability of the BFFDDCF generated by the generator is later computed by the tester. It does so by testing each of the packets from a sample in the BFFDDCF and checking whether it results in a certain decision or multiple decisions. Based on the results for all the packets in the sample, it then computes the BFFDDCF multiple decision probability.

The experiment was conducted based on security policies from two different firewalls, the first consisting of a small number of ACEs and the second consisting of a large number of ACEs, referred to as 'Firewall 1' and 'Firewall 2', respectively. Moreover, a few assumptions regarding statistical traffic parameters were made. A good packet ratio of 0.1 was assumed in all cases, and six different values of the rate of traffic at the public cloud entrance were considered. Based on the above assumptions, the experiment was conducted on automatically generated samples consisting of 3 932 160 and 62 914 560 network packets, respectively. The results are given in Table 1. The cells contain both the BFFDDCF multiple decision probability determined by the generator and the BFFDDCF multiple decision probability computed by the tester for each of the two firewalls and for each of the six values of the rate of traffic at the public cloud entrance. To better illustrate data from Table 1, these are shown as a plot in Fig. 12.

Based on the experimental results from Table 1 and Fig. 12, it can be seen that in each case the value of the BFFDDCF multiple decision probability computed by the tester is close to the value of the BFFDDCF multiple decision probability determined by the generator. Also, in most of the cases, the value computed by the tester is lower than the value determined by the generator. A few cases in which the value computed by the tester is higher than the value determined by the generator may be the result of a library used to implement the BF [22], a computational error, or the internal structure of the security policy. In most cases, the experimental results coincide with the theoretical values determined based on Formula 15 and shown in Fig. 9.

**Fig. 12** Experimental results—BF parameters selection

All operations were performed on a virtual machine running the CentOS 6.5 OS with a 2.6GHz processor and 4 GB of *Random-Access Memory* (RAM). In the course of the experiment, it turned out that the operation which is the most computationally expensive in the BFFDDCF generation process is the operation which transforms the BFFDD to the BFFDDCF. It was noticed that a relation of the BFFDDCF generation time to the number of combinations of good packets is linear and is around 1 s per 100,000. The experiment demonstrated that it is possible to design BFFDDCF in which its multiple decision probability is as close as possible to the desired value by selecting appropriate parameters of the BF's that build it up.

## 7 Limitations and future work

While it has been shown that the Ladon Hybrid Cloud is an effective framework for preserving cloud-based firewall policy confidentiality, it is important to mention its limitations and suggest directions for future work.

The first is a concern regarding the return traffic. In Sect. 3, it was stated that the main drawback of the Ladon framework—the risk of firewall deanonymization by packet eavesdropping and analysis—can be eliminated by introducing the purposefulness of packet decision uncertainty in the public cloud. Although this is true for the forward traffic, such information can still be gained by packets eavesdropping and performing analysis of the return traffic. This limitation applies to the *Transmission Control Protocol* (TCP) only. This is because the TCP requires all packets to be acknowledged so the packets permitted on the customer side will result in an acknowledgment in the return traffic. Despite the fact that this limitation raises a concern, a target solution is left for future consideration. However, three potential solutions are proposed below.

The first solution is to route the return traffic via another link which does not pass through the public cloud on the CSP side. As a result, the CSP is simply deprived of information regarding the return traffic. Another solution is to encrypt the return traffic. Although the forward traffic needs to pass unencrypted via the BFFDD on the CSP side, the return traffic may be encrypted to ensure that no information, including acknowledgments, can be retrieved. The third potential solution is to extend the return traffic with fake packets including fake acknowledgments. As a result, the knowledge base regarding the acknowledged packets built by the hostile CSP becomes inaccurate and expands over time.

Another limitation is the dependence of the Ladon Hybrid Cloud on statistical traffic parameters. In Sect. 4, it was mentioned that both the rate of traffic at the public cloud entrance and the good packet ratio values change over time. To address this, some constant values have been used instead which can be calculated based on the statistical traffic parameters. However, assuming that computational power will continue to increase, it is likely that in the future it will be possible to adjust the BFFDDCF multiple decision probability continuously based on real-time traffic analysis. This is because BFFDDCF computation is time-consuming even for modern, powerful computers. The above may also be a limitation for organizations which change their firewall policy frequently.

An interesting direction for future development of the Ladon Hybrid Cloud is analysis of the Ladon Hybrid Cloud security level in architectures with a federation of multiple clouds. In this article, just one public and one private cloud were assumed. However, as cloud federation technologies are becoming increasingly popular, it is possible to implement the Ladon Hybrid Cloud in one of the following architectures: many public clouds and one private cloud, one public cloud and many private clouds, and many public clouds and many private clouds. With fewer packets flowing between the public and private clouds, the CSP is able to obtain less information regarding the original firewall security policy; therefore, the expected Ladon Hybrid Cloud security level is higher in architectures with a cloud federation. Detailed analysis of such frameworks, required in order to demonstrate the validity of the assumptions, should further be performed.

## 8 Conclusions

The number of cloud-based services increases every year. The maturity of cloud computing technology encourages organizations to move subsequent types of services, previously impossible to outsource, into the cloud. This includes security services which include firewall services. However, those which have already begun to be widely adopted con-

tinue to suffer from information confidentiality and privacy issues as a result of firewall policy outsourcing.

While a framework, referred to as Ladon by its authors, preserving the confidentiality of the original firewall policy by introducing BFFDD has been proposed, it has a drawback: There is a risk of firewall deanonymization by traffic eavesdropping and analysis. To bypass this issue and limit the amount of information regarding the original firewall structure carried in packet headers, a novel framework introducing the purposefulness of packet decision uncertainty has been proposed in this article as an extension to Ladon. This extension known as the Ladon Hybrid Cloud leverages a hybrid cloud model and performs additional filtering of packets resulting in multiple decisions after passing through BFFDD in a private cloud on customer premises. Additional optimization techniques which help minimize the amount of information carried by particular packets based on the firewall policy type in use have also been proposed.

As computational resources of the private cloud are usually limited, an analysis of the Ladon Hybrid Cloud has been performed to check how the framework deals with this. It has been shown in the results of the analysis and confirmed in the results of the experiment that it is possible to control the rate of traffic at the private cloud entrance by selecting appropriate values of BF parameters while knowing basic traffic statistics. It has also been demonstrated that it is possible to find a trade-off between the Ladon Hybrid Cloud privacy level, its congestion probability, and efficiency.

The Ladon Hybrid Cloud allows organizations to take back control of privacy by helping them preserve their firewall policy confidentiality when outsourcing firewall services into the cloud. It extends the regular Ladon framework by eliminating its main drawback—the risk of firewall deanonymization by packets eavesdropping and analysis. The Ladon Hybrid Cloud is the final missing part of the puzzle which resolves the key issue of cloud-based firewall services: information confidentiality and privacy.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

## References

1. Furth, B.: Cloud computing fundamentals. In: Furth, B., Escalante, A. (eds.) *Handbook of Cloud Computing*, pp. 3–20. Springer, New York (2010)
2. SecaaS Working Group: Defined Categories of Service (2011). Cloud Security Alliance (CSA). <https://cloudsecurityalliance.org/>

- [wp-content/uploads/2011/09/SecaaS\\_V1\\_0](#) (2011). Accessed 1 Aug 2014
3. AT&T Intellectual Property: Managed Firewall Service Network-Based. AT&T, Inc. <http://www.business.att.com/content/productbrochures/Network-Based-Firewall> (2014). Accessed 1 Aug 2014
  4. Virtela Inc: Virtela Enterprise Services Cloud (ESC). Virtela, Inc. <http://www.virtela.net/services/virtela-esc/> (2013). Accessed 1 Oct 2013
  5. VMware Inc: Business and Financial Benefits of Virtualization. VMware, Inc. <http://www.vmware.com/files/pdf/cloud-journey/VMware-Business-Financial-Benefits-Virtualization-Whitepaper> (2011). Accessed 1 Aug (2014)
  6. Websense Inc: Seven Criteria for Evaluating Security-as-a-Service (SaaS) Solutions. Websense, Inc. <http://www.websense.com/assets/white-papers/whitepaper-seven-criteria-for-evaluation-security-as-a-service-solutions-en> (2010). Accessed 1 Aug 2014
  7. Zhou, M., Zhang, R., Xie, W., Qian, W. and Zhou, A.: Security and privacy in cloud computing: a survey. In: 6th International Conference on Semantics, Knowledge and Grids, pp. 105–112 (2010)
  8. Chen, D., Zhao, H.: Data security and privacy protection issues in cloud computing. In: International Conference on Computer Science and Electronics Engineering, pp. 647–651 (2012)
  9. Kurek, T., Lason, A., Niemiec, M.: First step towards preserving the privacy of cloud-based IDS security policies. *Secur. Comm. Netw.* 9999 (2015). doi:[10.1002/sec.1272](https://doi.org/10.1002/sec.1272)
  10. Trustwave SpiderLabs: 2012 Global Security Report. Trustwave Holdings, Inc. <http://www2.trustwave.com/rs/trustwave/images/2012-Global-Security-Report> (2012). Accessed 1 Oct 2013
  11. Trustwave SpiderLabs: 2013 Global Security Report. Trustwave Holdings, Inc. <http://www2.trustwave.com/rs/trustwave/images/2013-Global-Security-Report> (2013). Accessed 1 Aug 2014
  12. Scoop Media: Survey: 88 % of ICT Employees Would Steal. Scoop Top Stories. <http://www.scoop.co.nz/stories/BU0811/S00203.htm> (2008). Accessed 1 Aug 2014
  13. McAfee, R.B., Champagne, P.J.: Effectively Managing Troublesome Employees, Westport (1994)
  14. Khakpour, A.R., Liu, A.X.: First step toward cloud-based fire-walling. In: 31st International Symposium on Reliable Distributed Systems, pp. 41–50 (2012)
  15. Gouda, M.G., Liu, A.X.: Structured firewall design. *Comput. Netw. J.* 51(4), 1106–1120 (2007)
  16. Bloom, B.: Space/time trade-offs in hash coding with allowable errors. *Commun. ACM.* 13(7), 442–426 (1970)
  17. Jajodia, S., Ghost, A.K., Swarup, V., Wang, C., Sean Wang, X.: Moving Target Defense. Springer, New York (2011)
  18. Fulp, E.W., Tarsa, S.J.: Trie-based policy representations for network firewalls. In: 10th IEEE Symposium on Computers and Communications, pp. 434–441 (2005)
  19. Bose, P., Guo, H., Kranakis, E., Mahashwari, A., Morin, P., Morrison, J., Smid, M., Tang, Y.: On The False Positive Rate of Bloom Filters. Report, School of Computer Science. doi:[10.1016/j.ipl.2008.05.018](https://doi.org/10.1016/j.ipl.2008.05.018)
  20. EC-Council: The 5 phases every hacker must follow. EC-Council. <http://www.clrgroup.com/Site/pdf/EthicalHacking> (2013). Accessed 1 Oct 2014
  21. Jung, J., Krishnamurthy, B., Rabinovich, M.: Flash Crowds and Denial of service attacks: characterization and implications for CDNs and websites. In: Proceeding of 11th World Wide Web conference, pp. 293–304 (2002)
  22. Burstein, M.: Creating a simple Bloom filter. Max Burstein Blog. <http://maxburstein.com/blog/creating-a-simple-bloom-filter/>. Accessed 1 Aug 2014